SERIAL NO. 09/542,189

PATENT
Docket RAL920000008US1

CERTIFICATE OF MAILING (37 C.F.R. 1.8(a))
I hereby certify this correspondence is being deposited with the United States Postal service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on **JULY 1, 2004** ***,***
by Karen Orzechowski **Signature:** Karen Orzechowski

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

In re application of G. T. Davis, et al.	:	Date: July 1, 2004 IBM Corporation - IP Law 9CCA/B002 P.O. Box 12195
Serial No. 09/542,189	:	Research Triangle Park, North Carolina 27709
Filed: April 4, 2000	:	Customer Number 25299
For: NETWORK PROCESSOR WITH MULTIPLE INSTRUCTION THREADS	:	Unit: 2183
	:	Examiner: David J. Huisman

APPEAL BRIEF

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an appeal from the Final rejection of claims 1-31 of this application. An appendix containing a copy of the claims is attached.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation (IBM), the Assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

None of the claims have been canceled. Claims 1-31 are on appeal.

IV. STATUS OF AMENDMENT

An Amendment After Final (37 CFR 1.116) is filed concurrently with this Appeal Brief. The amendment addresses objections raised to the specification and claims 24, 28 and 30.

V. SUMMARY OF INVENTION

The present invention relates to a processor with multiple instruction threads and a controller for switching between threads based upon latency event associated with a task when execution of a thread stalls. For short latency event (e.g. delays of 25 machine cycles or less) the control is temporarily transferred to another thread and returned to the original thread upon expiration of the short latency event. For long

latency event (e.g. delays over 25 machine cycles) full control is permanently transferred to the other thread.

Access to data such as the data in a tree search structure is pipelined to enable the multiple execution threads to have overlapping search access. A first thread in the queue (Figure 3, FIFO 52) is executed until a latency event of a given duration occurs which causes the execution to stall. Control of the execution is then transferred to the second thread in the queue when execution of the first thread stalls. If the latency event is programmed to be short, e.g. 25 machine cycles or less, the control is temporarily transferred to the second thread. On the other hand, if the latency event is programmed to be longer, e.g. over 25 cycles, full control is transferred to the second thread in the queue. The duration of a typical machine cycle may be between about 5 and about 7.5 nanoseconds. The trend is toward ever faster cycles as technology continues to evolve. (Page 4, lines 4-12).

The invention also relates to a network processor (Figure 1) configuration comprising a CPU 10 with multiple threads, an instruction memory 16, and a prefetch queue 18 between the instruction memory and the CPU. An array of general purposes registers communicates with the CPU. The configuration includes local data storage and a thread execution control for the general register array and the local data storage. A first coprocessor connects the CPU to a local data storage. A pipelined coprocessor connects a shared remote storage and the CPU.

The invention includes a thread execution control (TEC) 30 Figure 2 and Figure 3 for the execution of multiple independent threads in a processor configuration. The TEC comprises a priority FIFO 52 to grant priority to one of a plurality of threads; an arbiter to control the execution of the prioritized threads, and a thread control state machine 38 for shifting execution control between threads upon the occurrence of latency events. The thread priority is granted by the FIFO by loading a thread number into FIFO when a

packet is dispatched to the processor. The thread number is unloaded from the FIFO when a packet has been enqueued for transmission. When a long latency event occurs, a thread number is circulated from highest priority to lowest priority in the FIFO. The thread outlets of the FIFO are used to determine priority depending on the length of time a thread has been in FIFO.

The thread execution control includes a thread control state machine (38 Figure 3). The machine grants control of execution from a first thread to a second thread when a latency event occurs in the execution of the first thread. The transfer is for temporary control to the second thread if the latency event causes a short latency stall of, e.g. less than about 25 CPU cycles. On the other hand, the transfer to the second thread is for full or permanent control if the latency event is a long latency event that causes a stall of more than about 25 cycles. Commonly, the threshold between a short latency and a long latency is decided by the microcode programmer and is encoded into the program.

The invention also comprises a system for granting execution priority to an independent thread based on the logical function of an arbiter based on the following Boolean expression:

$$G_n = R_n \cdot \{(P_A = n) + R_{PA} \cdot (P_B = n) + R_{PA} \cdot R_{PB} \cdot (P_C = n) \cdots\}$$

where: G_n is a grant from a given thread $_N$

R_n is a request from a given thread $_N$;

P_A, P_B and P_C represent a plurality of thread numbers depending on the number of threads in use and ranked by alphabetical subscript;

R_{PA} is a request from the highest priority thread in the FIFO; and

$_n$ is a subscript identifying a thread by the bit or binary number

The system determines whether a request R is active or inactive. It also determines the

priority of the threads and then matches the request R with the corresponding thread P. Finally, it grants a request for execution if the request is active and if the corresponding thread P has the highest priority.

The invention also involves the use of a prefetch buffer in connection with a plurality of independent thread processes in such a manner as to avoid an immediate stall when execution is granted to an idle thread. This involves determining whether the buffer is being utilized by an active execution thread. During periods that the buffer is not being used by the active execution thread, the buffer is enabled to prefetch instructions for an idle execution thread.

VI. ISSUES

A. Whether Claims 27, 29 and 30 are unpatentable under 35 USC 102(e) as being anticipated by Joy et al. (U.S. Patent No. 6,341,347 B1).

B. Whether Claims 1-7, 9-17 and 19-23 are unpatentable under 35 USC 103(a) over Joy et al. (U.S. Patent No. 6,341,347 B1) in view of Callon et al. (U.S. Patent No. 5,251,205).

C. Whether Claims 8 and 18 are unpatentable under 35 USC 103(a) over Joy et al. (U.S. Patent No. 6,341,347 B1) in view of Callon et al. (U.S. Patent No. 5,251,205) in view of Parady (U.S. Patent No. 5,933,627) and further in view of Horvitz (U.S. Patent No. 6,067,565).

D. Whether Claims 24 and 25 are unpatentable under 35 USC 103(a) over Joy

et al. (U.S. Patent No. 6,341,347 B1) and the American Heritage Dictionary of English Language Third Edition, 1992 defining que/queuing as used in Joy et al. in view of Parady (U.S. Patent No. 5,933,627).

E. Whether Claim 26 is unpatentable under 35 USC 103(a) over Joy et al. (U.S. Patent 6,341,347 B1) in view of Parady (U.S. Patent No. 5,933,627) in view of Lee (U.S. Patent No. 5,404,560) and further in view of Horwitz (U.S. Patent No. 6,067,565).

F. Whether Claims 28 and 31 are unpatentable under 35 USC 103(a) over Joy et al. (U.S. Patent No. 6,341,347 B1) in view of Nikhil et al. (U.S. Patent No. 5,499,349).

VII. GROUPING OF CLAIMS

There are six groups of claims.

Group 1 consists of Claims 27, 29 and 30. The claims of this group do not stand or fall together.

Group 2 consists of Claims 1-7, 9-17 and 19-23. The claims of this group do not stand or fall together.

Group 3 consists of Claims 8 and 18. The claims of this group do not stand or fall together.

Group 4 consists of Claims 24 and 25. The claims of this group do not stand or fall together.

Group 5 consists of Claim 26.

Group 6 consists of Claims 28 and 30. The claims of this group do not stand or fall together.

VIII ARGUMENTS

Issues and related arguments are identified by the same alphabetical characters.

A. The argument supporting patentability of Claims 27, 29 and 30 are as follows:

1. JOY ET AL. DOES NOT TEACH EVERY ELEMENT OF CLAIMS

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. Vordegaal Bros. v. Union Oil Co. of California, 814 F. 2d 628, 631, 2 USPQ 2d 1051, 1053 (Fed. Cir. 1987).

Applying this holding to the case at hand it is Appellants' contention elements a) and c) recited in Claim 27 are not found (expressly or inherently) in Joy et al. (U.S. Patent No. 6,341,347 B1). Claim 27 covers a thread execution control shown in Figure

3 and described in the specification. It seems as if Joy et al. uses an N-bit thread selectable flip flop logic circuit to control thread selection (see Figures 4a-4c and Figure 5; col. 10-col. 13, Joy et al.). The circuit in Joy et al. is different from the thread execution control recited in claim 27. As a consequence claim 27 is not anticipated by Joy et al.

Claim 29 depends on claim 27 and includes –by reason of its dependency– all the limitations of claim 27. As argued above and incorporated herein by reference elements of claim 27 not found in Joy et al. are also not formed in claim 29. Therefore, claim 29 is not anticipated.

In addition, claim 29 is separately patentable. The arbiter grants execution priority to an independent thread based upon the Boolean expression set forth in claim 29. This feature of granting thread priority based upon the recited Boolean function gives additional level of control as to which thread is granted priority and this feature is not found in Joy et al.

Claim 30 depends on claim 27 and is patentable over Joy et al. –by reason of its dependency– in that elements set forth above and incorporated herein by reference are not found in Joy et al.

In addition claim 30 is separately patentable in that it recites the control logic that drives the control state machine through its respective states. In Joy et al. no control logic for driving state machine is disclosed. Because this feature is not present (explicitly or inherently) in Joy et al. the claim is not anticipated.

The Examiner's contention that Joy et al. anticipates Claims 27, 29 and 30 is not supported by the record. In the first instance the Examiner's argument in support of the rejection appears circuitous and difficult to understand. It is Appellants' contention a rejection based upon anticipation is straightforward in that the claimed element is either

in the reference or it's not. The fact that the Examiner has to present such circuitous argument with what appears to be erroneous conclusion should be indication of the reference not anticipating claims 27, 29 and 30.

In addition, with respect to the rejection of claims 27, 29 and 30 the Examiner seems to rely on Joy et al. (U.S. Patent No. 6,341,347 B1) and the American Heritage Dictionary of the English Language, Third Edition. This seems to be reliance on two references which would violate the case law (see above) regarding a single reference to invalidate claims under 35 USC 102(e).

The Examiner's assertion that applicants' FIFO is not a FIFO (see page 4, lines 9-20, Final Rejection, Paper 9) is in error. The Examiner relied upon page 12, lines 20 to page 13, line 7, appellants' specification, to support this assertion. A review of the paragraph indicates nothing that would support the Examiner's conclusion. Moreover words, terms etc. in a claim are given their ordinary meaning. Resort to the specification is only made if the claimed terms, words etc. are ambiguous. It is applicants' contention there is no ambiguity in the claim language "a priority FIFO buffer for storing thread numbers". Therefore, the Examiner's reliance on the specification to argue otherwise appears to be error.

It is noted there is no mention of FIFO in Joy et al. much less a FIFO to be used as set forth in the claims. As a consequence this element of claims 27, 29 and 30 is not found in Joy et al. and the claims are not anticipated.

Likewise the Examiner's statement that "an arbiter for determining . . . " (element c) of the claims is inherently disclosed in Joy et al. is in error; since there are other ways, as stated in Joy et al., of making a thread selection without use of the arbiter as set forth in appellants' claims. Therefore, the element (c) is not found in the references.

Regarding what may be "inherent", however, the Federal Circuit has set out clear

standards for such a showing. To establish inherency the extrinsic evidence must make clear that the missing descriptive matter is necessarily present in the thing described in the reference and that it would be so recognized by persons of ordinary skill. In re Robertson, 169 F.3d 743, 745, 49 USPQ 1949, 1950-51 (Fed. Cir. 1999) (other citations omitted). It is appellants' contention the Examiner has not provided any extrinsic evidence that meets the requirement set forth in In re Robertson. Absent evidence in support of the finding by the Examiner that is disputed by appellants, the disputed element (c) is not disclosed (inherently or otherwise) in Joy et al. Therefore, claims 27, 29 and 30 are not anticipated.

With respect to claim 29, the Examiner's assertion " . . . Joy has further taught that the arbiter controls the priority of execution of multiple independent threads based on the boolean expression recited in claim 29" is in error. The Boolean expression of claim 29 includes

$$G_n = R_n \cdot \{(P_A = n) + R_{PA} \cdot (P_B = n) + R_{PA} \cdot R_{PB} \cdot (P_C = n) \cdots\}$$

where: G is a grant

R_n is a request from a given thread;

P_A, P_B and P_C represent threads ranked by alphabetical subscript according to priority;

$_n$ is a subscript identifying a thread by the bit or binary number

The Joy et al. reference does not even mention the use of a Boolean expression much less the one set forth above. As a consequence claim 29 not anticipated.

Finally, appellants respectfully disagree with the Examiner's finding that the operation of appellants' "Control State Machine" set forth in claim 30 is found in Joy et

al. Figure 3 shows block 310 with the label "Thread 0 Machine State" and similar reference at col. 8, line 28.

It is appellants' contention this reference in Joy et al. is insufficient to conclude that the reference (Block 310) in Joy et al. meet the recitation of claim 30 because a state machine can operate in a plurality of different ways none of which includes the way set forth in claim 30. Moreover, the Examiner has not demonstrated the recitation of claim 30 is inherent in Joy et al. The Examiner is obligated to make such a showing and he has not done so. As a consequence claim 30 is not anticipated.

B. The issue presented is whether claims 1-7, 9-17 and 19-23 are unpatentable under 35 USC 103(a) over Joy et al. (U.S. Patent 6,341,347 B1) in view of Callon et al. (U.S. Patent No. 5,251,205). For reasons set forth herein appellants contend the claims are not obvious.

b) 1. NO SUGGESTION OR MOTIVATION TO COMBINE REFERENCES

Joy et al. (U.S. Patent No. 6,341,347 B1) describes a multi-threaded processor 300 (Figure 3) that switches from one thread to another upon the occurrence of a stall condition in the thread that is currently being executed. The switch occurs in response to L1 cache mis and interrupts (col. 16, lines 9-20).

Callon et al. (U.S. Patent No. 5,251,205) discloses tree structure that identify routes within a network of routers.

The subject matter of claim 1 and dependent claims (2-7 and 9) relates to using multi threads in association with network processor and accessible data available in a

tree search structure. In particular, the claims, among other elements, call for "queuing the multiple execution threads to have overlapping access to the accessible data available in said tree search structure".

It is appellants' contention Joy et al. does not teach the recited claimed element[1] and combining Callon et al. due to Callon et al. discloses tree structure is an improper combination. The combination is improper because there is no suggestion or motivation to combine in any of the references. Absent such motivation or suggestion and failure of the Examiner to set forth logical and concrete reasons why an artisan viewing the references, without hindsight of appellants' disclosure, would form the combination raises an inference or suspicion that the basis for the combination is appellants' own disclosure. An examiner cannot arbitrarily pick and choose elements from the prior art in a piecemeal fashion to construct the claimed invention without some direction from the prior art. See In re Donovan, 509 F.2d 554, 184 USPQ 414 (CPPA 1975). Where the art does not suggest the desirability of a combination, the combination is not obvious. In re Imperato, 486 F.2d 585, 179 USPQ 730 (CCPA 1973).

Reliance on appellants' disclosure is contrary to the requirements of 35 USC 103. As the court stated in In re Sponnoble, 160 USPQ 237, 243 (CCPA 1969): "The court must be ever alert not to read obviousness into an invention on the basis of the applicant's own statements; that is, we must view the prior art without reading into that art appellant's teachings. In re Murray, 46 CCPA 905, 268 F 2d 226, 122 USPQ 364; In re Sporck, 49 CCPA 1039, 301 F. 2d 686, 133 USPQ 360. The issue then is whether the teachings of the prior art would, in and of themselves and without the benefits of

[1] It seems as if the Examiner does not disagree (see penultimate sentence, page 7 final Office Action).

applicant's disclosure, make the invention as a whole, obvious. In re Leonor, 55 CCPA 1198, 395 F. 2d 801, 158 USPQ 20." (Emphasis Court)

This improper combination is also censored by the Board of Appeals in Ex party Fleischmann, 157 USPQ 155 (1967). In that case, the examiner found it necessary to combine four references to reject certain claims under 35 USC 103, and a fifth reference to likewise reject another claim. The Board states its objection as follows: "After having reviewed the stated rejection in the light of the prior art and considered the respective positions taken by the examiner and appellant, we have concluded that the combination of references as applied by the examiner is unwarranted.

While as an abstract proposition, it might be possible to select features from the secondary references, as the examiner has done, and mechanically combine them with the Mallin device to arrive at appellant's claimed combination, we find absolutely no basis for making such combination neither disclosed nor suggested in the **patents relied** upon. In view only appellant's specification suggests any reasons for combining the features of the secondary references with the primary reference and under the provisions of 35 USC 103 that does not constitute a bar "

The Board's position is consistent with other Courts. In Simmonds Precision Products Inc. v United States, 153 USPQ 465 1967), the Court of Claims quoted the U.S. Supreme Court' in Graham v John Deere Co. 248 USPQ 459 (1966) as defining several factual inquiries useful in resolving the question of obviousness under 35 USC 103, namely: "Under Sec. 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved."

The Court of Claims then rejected an "attempted combination of ...Smithror Schafete et al....with certain aspects of the Laboulais control system to meet the recital

of element and function in Eddleman Claims" --- because such combination "requires hindsight view of the prior art." The Court then stated the following: "There is a temptation to read into the **prior art the teachings of the Eddleman** patent relative to avoiding the **effect of the dielectric** constant. However, prior disclosures should not **be combined to show** obviousness when there is no suggestion in **any of the disclosures** that the separate concept can be combined to produce the Eddleman gauge results.

The Rule of Law stating that numerous prior art **references** combined by the Examiner must themselves suggest the Examiner's combination (rather than allowing the Examiner to rely on the Appellant's disclosure with hindsight in producing his combination) is based upon the fact that it is the combination and not the individual elements of the combination, **that** comprises the claimed invention.

Therefore, in In re Avery, 186 USPQ 161 (1975), the CCPA refused to accept the combination of two patents to invalidate certain claims. The Court's decision was based on the fact that the need to use a feature of the secondary reference was not present in the primary Magee et al reference. The Court states its position as follows: "Because Magee et al has no use for an inert gas in their process, we do not believe that it would be obvious to combine the disclosure of Magee et al and Parry. neither reference contains the slightest suggestion to what it discloses in combination with what is disclosed in the other."

In In re Skole, 187 USPQ 481 (1975) the CCPA reaffirms its previous holding of In re Schaffer, 108 USPQ 326 (1956) wherein the Court states: "References may be combined for the purpose of showing that a claim is unpatentable. However, they may not be combined indiscriminately, and to determine whether the **combination** of references is proper, the following criterion is often used: namely, whether the prior art

suggests doing **what an applicant has** done ***. It is not enough for a valid rejection to view the prior art in retrospect once an applicant's disclosure is known. The art applied should be viewed by itself to see if it fairly disclosed doing what an applicant has done."

Finally, the Board of Appeals has censored the improper combination in Ex party Clapp, 227 USPQ 972 (1985). In that case the examiner combined several references to reject all the claims under 35 USC 103. In reversing the examiner, the Board states its position as follows: "Presuming arguendo that the reference shows the elements or concepts urged by the examiner, the examiner has presented no line of reasoning, and we know of none, as to why the artisan viewing only the collective teachings of the references would have found it obvious to selectively pick and choose various elements and/or concepts from the several references relied on to arrive at the claimed invention.*** To support the conclusion that the claimed combination is directed to obvious subject matter, either~the references must expressly or impliedly suggest the claimed combination or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the reference.

It is appellants' position the references do not suggest any motivation for the combination. Moreover, there is no need to use the tree structure of Callon et al. with Joy et al. (the primary reference) especially since Joy et al. switching is in conjunction with L1 cache used for information storage whereas tree structures are usually associated with routing of data as opposed to data storage. In addition, the examiner has not given any logical and concrete reasons for the combination. Therefore, the claims are unobvious.

Joy et al. (U.S. Patent No. 6,341,347 B1) and Callon et al. (U.S. Patent No. 5,251,205) do not suggest nor set forth any motivation for the combination. Regarding

claim 1 the only reason set forth by the Examiner is that it would be obvious to modify the system of Joy to incorporate a tree data structure in order to achieve more efficient data operations.

It is appellants' contention this reason is not concrete or sufficient to justify the combination. As a consequence the Examiner has not provided a prima facie case of obviousness. Therefore, the claims of this group are patentable over the art of record.

2. EXAMINER'S COMBINATION DOES NOT SUGGEST APPELLANTS' CLAIMS

It is appellants' contention that the resulting combination formed by the Examiner does not render the claims obvious. Claim 1, element (b), in part, states: "queueing the multiple execution threads to have overlapping access . . . "

None of the prior art references teaches this feature of appellants' claim. Joy et al. (main reference) teaches a single_processor verticaly_threaded processor 300 having a single pipeline shared among a plurality of machine states or thread, holding a plurality of machine states concurrently (Figure 3, col. 8, lines 15-26). The Joy et al. describes several variants of this basic structure. But nowhere in the Joy et al. reference is a teaching or suggestion of "queueing the multiple execution threads to have overlapping access - as is claimed by appellants. By so doing appellants provide a system that is better able to handle the rapid flow of data provided by the tree structure.

It is appellants' contention because this feature of appellants' claims is missing from the Examiner's combination the claims of this group are not obvious.

3. JOY ET AL. TEACHES AWAY FROM CLAIMED INVENTION

Joy et al. teaches thread switching in connection with L1 cache miss, instruction buffer empty and interrupts. (Joy et al. col. 16, lines 9-20). The event with which Joy et al. is associated represents non-availability of data.

Claim element (b) states, "queuing the multiple execution threads to have overlapping access to the accessible data available in said tree search structure". Based upon this element appellants contend the event with which this group of claims relates is "availability" of data. The "availability" of data and "non-availability" of data are inapposite and one does not suggest the other. In fact the teachings of Joy et al. as it relates to multi thread switching and non-availability of date event would not suggest multi-thread switching and availability of data events to an artisan. In this regard the teachings in Joy et al. appears to be inapposite to subject matter claimed in this group. Therefore, the claims are not obvious.

4. Furthermore, appellants contend the association of thread switching with data from tree search structure (claim 1, element b) provide a new use for an old technique (i.e. thread switching on stalled thread). This, appellants assert, represents evidence of unobviousness.

5. PATENTABILITY OF CLAIM 2

Claim 2, due to its dependency on claim 1, is patentable over the art of record for reasons set forth above and incorporated herein by reference.

In addition claim 2 is separately patentable. The claim calls for the control of the execution to be temporarily transferred to the next thread when execution stalls due to short latency event, and the control is returned to the original thread when the event is completed. A short latency event is defined as 25 machine cycles or less (see appellants' specification page 4, lines 8-12; page 7, lines 19-21 through page 8, lines 1-9). By temporarily switching controls, as set forth in claim 2, system performance is enhanced (see page 8, lines 1-9). The enhancement of system performance provided by this feature is a benefit. In contrast, "temporary switching of control" is not disclosed in Joy et al. As a consequence this feature is novel. It is appellants' contention novel process step and benefits are indicia of non-obviousness.

6. PATENTABILITY OF CLAIM 3

Claim 3 is patentable –due to its dependency– for reasons set forth above and incorporated herein by reference.

In addition, claim 3 is separately patentable for reciting "a processor instruction is encoded to select a short latency event". Joy et al. does not teach this feature. Therefore it is novel. The feature further enhances system performance which is a benefit. Novel process step and benefits are indicia of non-obviousness.

7. PATENTABILITY OF CLAIM 4

Claim 4, due to its dependency on claim 1, is patentable for reasons set forth

above and incorporated herein by reference.

Claim 4 is separately patentable in that it calls for full transfer of execution to be given to next thread when execution of the first thread is stalled due to long latency. Long latency is defined as greater than 25 machine cycles (see appellants' specification page 4, lines 8-12; page 7, lines 19-21 through page 8, lines 1-9). Joy et al. does switch from one thread to the next when a stall occurs; but switching is not done due to long latency event as set forth in claim 4. Therefore, this feature of claim 4 is novel. The benefits derived from this novel feature are set forth at page 8, lines 5-9, appellants' specification. Appellants contend novel process step and benefits are indicia of non-obviousness.

8. PATENTABILITY OF CLAIM 5

Claim 5, due to its dependency, is patentable for reasons set forth above and incorporated herein by reference.

Claim 5 is separately patentable. It calls for a process or instruction to be encoded to select a long latency event. No such encoding is done at Joy et al. Therefore, the process is novel. The feature further enhances system performance which is a benefit. As pointed out herein novel process steps and benefits are indicia of nonobviousness.

9. PATENTABILITY OF CLAIM 6

Claim 6 stands or falls with claim 1.

10. PATENTABILITY OF CLAIM 7

Claim 7 –by reason of its dependency on claim 1– is patentable for reasons set forth above and incorporated herein by reference.

Claim 7 is separately patentable. It calls for the threads have overlapping access to shared remote storage via a pipelined coprocessor by operating within different phases of a pipeline of the coprocessor. Joy et al. does not disclose the feature. Therefore, it is novel. The feature provides enhancement to system throughput in that the main processor is forced to perform task while the co-processor, during all phases of operation, coacts with remote memory. The enhancement and novel feature are evidence of nonobviousness.

11. PATENTABILITY OF CLAIMS 9 and 11

Claims 9 and 11 stand or fall with claim 1.

12. PATENTABILITY OF CLAIM 12

Claim 12 stands or falls with claim 2.

13. PATENTABILITY OF CLAIM 13

Claim 13 stands or falls with claim 3.

14. PATENTABILITY OF CLAIM 14

Claim 14 stands or falls with claim 4.

15. PATENTABILITY OF CLAIM 15

Claim 15 stands or falls with claim 5.

16. PATENTABILITY OF CLAIM 16

Claim 16 stands or falls with claim 6.

17. PATENTABILITY OF CLAIM 17

Claim 17 stands or falls with claim 7.

18. PATENTABILITY OF CLAIM 19

Claim 19 stands or falls with claim 9.

19. PATENTABILITY OF CLAIM 20

Claim 20 stands or falls with claim 19.

20. PATENTABILITY OF CLAIM 21

Claim 21 is separately patentable. The claim calls for "the general purpose registers and local data storage are made available to the processor by providing one address bit under the control of the thread execution control logic and by providing the remaining address bits under the control of the processor.

By providing and structuring the address bits as recited in the claim the thread execution control logic can determine which part of the register array is used by a thread and the CPU can access registers associated with a particular thread (see page 9, lines 2-13, appellants' specification). The benefit is that this function (including bit assignment) which is not found in Joy et al. enhances the capability of appellants' system to switch more efficiently between threads for short latency events or long latency event. It is appellants' contention novel structure and benefits are evidence of nonobviousness.

21. PATENTABILITY OF CLAIM 22

Claim 22, due to its dependency on claim 20, is patentable for reasons set forth above and incorporated herein by reference.

In addition claim 22 is separately patentable. Claim 22 calls for the processor to be capable of simultaneously addressing multiple register array and a selector selects which array to be delivered for a given thread. By so doing the control logic is made less complicted since the only decision in this regard is to determine which array to assign to a thread. It also enhances the switchover capabilities of the system with zero overhead CPU cycles. (See appellants' specification, page 14, lines 21 through page 15, lines 1-15).

The processor being capable of addressing multiple array is a novel structure

and contribution to switch over with zero overhead is a benefit. The novel structure and benefits are evidence of non-obviousness.

Furthermore appellants argue the reference appears to teach away from claim 22. Joy et al. Figure 13 and col. 26, lines 42-68 and col. 27, lines 1-52 show and describe a multi-dimensional register file. The patent does not teach the processor has capability to address multiple array as is set forth in claim 22. However, Joy et al. seems to suggest simultaneous activation of multiple planes or windows is prohibited. In fact only one window is activated at any instant of time (see col. 27 lines 26-37). If only one window is activated this implies the processor only needs to address the single plane and not multiple ones as recited in claim 22. Therefore, the reference teaches away from claim 22.

22. PATENTABILITY OF CLAIM 23

Claim 23 is separately patentable. It allows local storage to be fully addressable by the processor, and index register is contained within the register array and the thread execution control has no address control over the local data storage or register array. This has the advantages of enabling some shared memory to be used for common data, such as tables (see appellants' specification page 15, lines 11-15). Appellants contend the novel structure and advantage not recognized in Joy et al. are evidence of nonobviousness.

C. The issue presented is whether Claims 8 and 18 are patentable, under 35 USC 103(a), over Joy et al. (U.S. Patent No. 6,341,347) in view of Callon (U.S. Patent No. 5,251,205) in view of Parady (U.S. Patent No. 5,933,627) in view of Horvitz (U.S.

Patent No. 6,067,565).

23. PATENTABILITY OF CLAIM 8

Claim 8 depends on claim 1 and due to its dependency is patentable over the cited references for reasons set forth above and incorporated by reference.

In particular the argument set forth above regarding the improper combination of Joy et al. and Callon is equally applicable and incorporated herein by reference. The reasons set forth above for the improper combination are not cured by Parady and/or Horvitz references. In this regard the cited references are merely cumulative.

In addition claim 8 is separately patentable. Among other things (set forth therein) claim 8 calls for " . . . collecting instructions in a prefetch buffer for its execution thread when the thread is idle and when the instruction bandwidth is not being fully utilized". Under the condition of claim 8 the system CPU is still operational and not idle.

Horvitz (U.S. Patent No. 6,067,565 teaches the web browser prefetching web page from the Internet during idle CPU or network capacity that would otherwise be wasted (col. 3, lines 25-56, Horvitz).

It is appellants' contention the condition for prefetching in Horvitz is different and inapposite to the condition for prefetching instruction in claim 8. Therefore, an artisan viewing the teachings in the Examiner's combination would be lead away from forming a combination that would render claim 8 obvious. Likewise, the Examiner's combination would lack the conditions (set forth above) required by claim 8. As a consequence even after the combination the resulting reference would not render claim 8 obvious.

23(i) TEACHINGS OF HORVITZ (U.S. PATENT 6,067,565) APPEARS NON-ANALOGOUS TO CLAIM 8

Claim 8 covers providing separate prefetch instruction buffer for each execution thread and collecting instructions in a prefetch buffer for its execution thread when the thread is idle and when the instruction bandwidth is not being fully utilized.

Horvitz, U.S. PATENT 6,067,565, relates to prefetching, from the Internet, web page of potential future interest in lieu of continuing current information download.

Appellants contend prefetching web pages from the Internet is a different technology from managing instruction in a processor. As a consequence, an artisan viewing the teachings of Horvitz is less likely to use it in a way that would render claim 18 obvious. Therefore, claim 8 is patentable over the art of record.

24. PATENTABILITY OF CLAIM 18

Claim 18 depends on claim 11 and is patentable for reasons set forth therein and incorporated herein by reference.

In addition claim 18 is separately patentable. Claim 18 relates to claim 8 in that claim 18 covers the processing system to practice the method of claim 8. As a consequence the arguments set forth in 23 are equally applicable to claim 18 and are incorporated herein by reference.

D. The issue presented is whether claims 24 and 25 are unpatentable, under 35 USC 103(a), over Joy et al (U.S. Patent No. 6,341,347) in view of Parady (U.S. Patent No. 5,933,627) and The American Heritage

Dictionary of English Language, Third Edition, used to define queue/queuing as used in Joy et al.

25. PATENTABILITY OF CLAIM 24

Claim 24 calls for (a) . . . an instruction memory and separate prefetch que for each thread between the instruction memory and the CPU . . . ; (b) a thread execution control . . . including priority FIFO buffer to store thread numbers. Neither (a) nor (b) is suggested in Joy et al. and/or Parady.

It is appellants' contention the structure of (a) is not shown or suggested in any of the Examiner's references. Therefore, the structure is novel. The benefit from the novel structure is the effective bandwidth for instruction fetching from an instruction memory is increased significantly with multiple execution threads (see appellants' specification page 15, lines 14-21; page 16, lines 1-8). The novel structure with benefits not recognized in either of the references is evidence of nonobviousness.

Regarding (b) FIFO buffer it is not disclosed or suggested in the references. By implementing the FIFO as part of the Thread Execution Control (TEC) the logic is made simpler requiring less circuit, a benefit to users. Therefore, novel structure and benefits indicate the claim is unobvious.

26. PATENTABILITY OF CLAIM 25

Claim 25, by reason of dependency on claim 24, is patentable over the art of record for reasons set forth under 24 above and incorporated herein by reference.

In addition claim 25 is separately patentable. Claim 25 covers portion of the

thread execution logic including an arbitor responsive to signals from the FIFO buffer and the state machine to determine thread execution. This structure is novel and not shown in Joy et al. By using this structure the complexity of circuits to manage threads of a multi-thread system is reduced. The reduction in circuit complexity is deemed as a benefit to users. The novel structure coupled with benefits are indicia of nonobviousness.

In reviewing the Joy et al. reference appellants found it very difficult to identify the thread execution control logic. A review of the Office Action did not find specific reference to drawings and/or teaching disclosing a specific design. Such identification would be helpful in determining whether the teachings in Joy et al. render claim 25 obvious.

It seems as if the only teachings of thread execution control logic in Joy et al. is shown in Figures 5 and 6 and described on col. 13, lines 5-67 through col. 14, lines 1-35. However, Figures 5 and 6 are only schematic blocks not showing any details that could be used to ascertain a design as set forth in appellants' claim 25. As a consequence the structure and benefits of claim 25 is not suggested in Joy et al. Therefore, claim 25 is patentable.

E. The issue presented is whether claim 26 is unpatentable, under 35 USC 103(a), over Joy et al. (U.S. Patent No. 6,341,347) in view of Parady (U.S. Patent No. 5,933,627) in view of Lee et al. (U.S. Patent No. 5,404,560) and further in view of Horvitz (U.S. Patent No. 6,067,565).

27. IMPROPER COMBINATION REFERENCES DO NOT SUGGEST MOTIVATION FOR COMBINATION

The Law as it relates to a rejection under 35 USC 103(a) based upon combination of references is set forth under B above and incorporated herein by reference. Simply put a rejection under 35 USC 103(a) based upon combination of references requires motivation for the combination to be found in at least one of the references or the Examiner sets forth logical and concrete reasons why an artisan viewing the references, without hindsight of appellants' disclosure, would form the combination.

Applying these principles to claim 26 and cited references appellants contend no motivation is found in any of the references. Therefore, the claim is not obvious. Claim 26 covers use of prefetch buffer together with multi thread to fetch instructions when the thread is not active. The claim set forth four elements (a - d). The Examiner cited four references set forth above in this section and seems to admit the claimed elements are to be found in respective ones of the references.

As shown below, appellants respectfully disagree that the claimed elements are shown or suggested in the references. But even if they were shown appellants contend no motivation for the combination could be found in any of the references. Each of the references set forth complete structure and teachings to support respective invention. Therefore, there is no need or reason to combine elements.

Even where the reference fails to suggest motivation to combine the combination could still be proper if the Examiner sets forth logical and concrete reason for the combination. It is appellants' contention the Examiner has not done so. For example, in combining Parady with Joy et al. the Examiner relied on teachings in Parady at column 3, lines 51-56 and conclude it would have been obvious to one of ordinary skill in the art at the time of the invention to provide a prefetch buffer for each thread. See page 19, section 41 of Final Office Action. It is appellants' contention the above is a mere

conclusion and does not provide concrete reason or motivation for the combination. Similar defects are also present in the other arguments in supporting combining elements from other references.

As a consequence the Examiner has not met his burden supporting a prima facie case of obviousness. Therefore, claim 26 is not obvious.

28. HORVITZ (U.S. PATENT 6,067,565) APPEARS TO BE NON-ANALOGOUS ART

Claim 26, element (d) calls for prefetching instruction in buffer when buffer is not being used by an active execution thread.

The Examiner relied on Horvitz for teaching this feature. Appellants believe reliance is in error since Horvitz relates to a different field of technology. Horvitz teaches prefetching from the Internet a web page of potential future interest during CPU inactivity. Because this teaching is different from claim 26 element (d) an artisan would not look to this piece of art to form a combination that would render claim 26 obvious. Therefore, claim 26 is not obvious.

29. REFERENCES DO NOT TEACH CLAIM 26 ELEMENTS a), c, and d)

Claim 26 is a method comprising a) associating each thread with a prefetch buffer. Parady (U.S. Patent No. 5,933,627) teaches instruction buffers 154, one for each of threads 0-3 (col. 5, lines 9-10). The claimed element calls for "prefetch buffer" and not "buffer" stated in the reference. In Parady no prefetch activity is associated with

the buffers. Therefore, this act of the method claim is not found in the reference.

Claim 26 c) states, in part, "determining whether the thread associated with the buffer is active. This feature is not found in Parady or any of the other references. Parady teaches switching between threads and upon a thread switch, the stream of instruction in one instruction buffer will simply pick up where it left off (Parady, col. 3, lines 50-55). This teaching relates to instruction processing and can hardly be construed as teaching or suggesting claim 26 (c).

Finally, claim 26 d) is not found in any of the references. Claim 26 d) states, in part, "during periods that the buffer is not being used by an active execution thread, enabling the buffer to prefetch instructions for the execution thread.

As argued above and incorporated herein by reference Horvitz (U.S. Patent No. 6,067,565) relates to prefetching web page from the Internet during CPU inactivity. This teaching is different from claim 26 d).

In view of the above, claim 26 is novel. Prefetching instruction in the dedicated instruction buffer when it is not being used by the associated thread increases efficiency of the system, and is therefore, a benefit.

It is appellants' contention novel process steps and benefits are indicia of nonobviousness.

In addition appellants argue that even after the combination the resulting method would not render claim 26 obvious because the acts set forth above would not be present. Without them being suggested in the references the claim is not obvious.

F. The issue presented is whether claims 28 and 31 are unpatentable under 35 USC 103(a), over Joy et al. (U.S. Patent No. 6,341,347) in view of Nikhil et al. (U.S. Patent No. 5,499,349).

30. PATENTABILITY OF CLAIM 28

Claim 28, due to its dependency on claim 27, is patentable for reasons set forth above and incorporated herein by reference.

In addition, claim 28 is separately patentable. The claim covers portion of the thread execution control including a FIFO buffer and means for performing the various functions recited in the claim relative to the buffer. As argues above and incorporated by reference the entire teachings in Joy et al. do not disclose the thread execution control of claim 28. Instead Joy et al. execution control appears to be the block diagrams in Figures 5 and 6 with associated description at col. 13, lines 1-64 and col. 15, lines 1-26. This showing in Joy et al. could never be construed to be suggestive of claim 28. The Nikhil et al. reference (U.S. Patent No. 5,499,349) does not provide the deficiency identified in Joy et al. Therefore, the Examiner's combination does not render claim 28 obvious.

Moreover, as argued above and incorporated herein by reference the Examiner's combination is improper since there is no motivation set forth in any of the references for the combination and the Examiner has not given any concrete and logical reason why an artisan viewing these references would ever form the combination suggested by the Examiner. It is settled law that hindsight based upon applicants' disclosure should not be used as the basis for forming the combination. For reasons set forth above it appears as if hindsight in forming the combination is present and as a consequence the claim 28 is patentable over the art of record.

30. PATENTABILITY OF CLAIM 31

Claim 31, due to its dependency on claim 28 is patentable for reasons set forth above and incorporated herein by reference.

In addition claim 31 is separately patentable. Claim 31 relates to the execution thread control and calls for means in the FIFO to detect occurrence of latency events. By so doing the efficiency of the execution thread control is maximized in that the thread is switched to another thread permanently or temporarily. Such combination of FIFO and means to detect latency event is not suggested in Joy et al. The combination also makes the design of the thread control simpler than it normally would have been. As a consequence the combination is novel and provides benefits. Therefore, claim 31 is patentable over the art of record.

APPELLANTS' RESPONSES TO ISSUES RAISED BY THE FINAL OFFICE ACTION

Regarding the rejection of claims 1-7, 9-17, and 19-23 (Issue B above) the Examiner states a) "It should be noted that such a tree structure is immaterial to the claimed inventive concept of thread switching and that the particulars of this structure in its claimed form have no effect on the operation of such a thread-switching system". b) "It is noted by the examiner that Joy does not teach the tree search structure. However, this structure has no effect on the inventive concept and is not given patentable weight in its claimed form".

"Tree Search Structure" is an integral feature recited in claim 1, element (b). Based upon (a) and (b) above the Examiner elects not to give "patentable weight" to it.

This, appellants contend is reversible error. The Examiner is obliged to examine the claims as presented. The Examiner may reject claims based upon statutory

grounds. The Examiner may even not give patentable weight to feature in the preamble of the claims. But it is believed the Examiner may not reconstruct claims by not giving patentable weight to features recited in the body of the claim as it is in claim 1 of this appeal. With respect to the present appeal this is of particular interest since examining the claim presented as a whole renders the prior art less relevant to the claims which includes "Tree Search Structure". As a consequence appellants respectfully petition the Board of Appeals to make a ruling on this issue.

In connection with combining references to reject claims under 35 USC 103(a) the Examiner cited In re Oetiker, 24 USPQ 2d 1443 (CAFC 1992). Appellants completely agree with the laws set forth in the cited case. But argue the Examiner's application of the law to appellants' application is in error and ought to be reversed by the Board. It is so requested .

CONCLUSION

Based upon the above arguments, the appealed claims define patentable subject matter over art of record. As a consequence the final rejection of claims 1-31 should be reversed.

Respectfully submitted,

Joscelyn G. Cockburn

Joscelyn G. Cockburn
Reg. No. 27,069
Attorney of Record

JGC:ko
Phone: 919-543-9036
FAX: 919-254-2649

Appendix of Rejected Claims:

1. The use of multiple threads in association with a network processor and accessible data available in a tree search structure, including the steps of:
   a) providing multiple instruction execution threads as independent processes in a sequential time frame;
   b) queuing the multiple execution threads to have overlapping access to the accessible data available in said tree search structure;
   c) executing a first thread in the queue; and
   d) transferring control of the execution to the next thread in the queue upon the occurrence of an event that causes execution of the first thread to stall.

2. The use of the multiple threads according to claim 1 wherein the control of the execution is temporarily transferred to the next thread when execution stalls due to a short latency event, and the control is returned to the original thread when the event is completed.

3. The use of multiple threads according to claim 2 wherein a processor instruction is encoded to select a short latency event.

4. The use of the multiple threads according to claim 1 wherein full control of the execution is transferred to the next thread when execution of the first thread stalls due to a long latency event.

5. The use of multiple threads according to claim 4 wherein a processor instruction is encoded to select a long latency event.

6. The use of multiple threads according to claim 1 including queuing the threads to provide rapid distribution of access to shared memory.

7. The use of the multiple threads according to claim 1 wherein the threads have overlapping access to shared remote storage via a pipelined coprocessor by operating within different phases of a pipeline of the coprocessor.

8. The use of the multiple threads according to claim 1 further including the step of providing a separate instruction pre-fetch buffer for each execution thread, and collecting instructions in a prefetch buffer for its execution thread when the thread is idle and when the instruction bandwidth is not being fully utilized.

9. The use of the multiple threads according to claim 1 wherein the threads are used with zero overhead to switch execution from one thread to the next.

10. The use of the multiple threads according to claim 9 wherein each thread is given access to general purpose registers and local data storage to enable switching with zero overhead.

11. A network processor that uses multiple threads to access data, including:
    a) a CPU configured with multiple instruction execution threads as independent processes in a sequential time frame;
    b) a thread execution control for
        1) queuing the multiple execution threads to have overlapping access

to the accessible data;

2) executing a first thread in the queue; and

3) transferring control of the execution to the next thread in the queue upon the occurrence of an event that causes execution of the first thread to stall.

12. A processing system utilizing multiple threads according to claim 11 wherein the thread execution control includes control logic for temporarily transferring the control to the next thread when execution stalls due to a short latency event, and for returning control to the original thread when the latency event is completed.

13. The processing system according to claim 12 wherein a processor instruction is encoded to select a short latency event.

14. The processing system according to claim 11 wherein the control transfer means includes the means for transferring full control of the execution to the next thread when execution of the first thread stalls due to a long latency event.

15. The processing system according to claim 14 wherein a processor instruction is encoded to select a long latency event.

16. The processing system according to claim 11 including means to queue the threads to provide rapid distribution of access to shared memory.

17. The processing system according to claim 16 wherein the threads have

overlapping access to shared remote storage via a pipelined coprocessor by operating within different phases of a pipeline of the coprocessor.

18. The processing system according to claim 11 further including a separate instruction pre-fetch buffer for each execution thread, and means for collecting instructions in a prefetch buffer for an idle execution thread when the instruction bandwidth is not being fully utilized.

19. The system according to claim 11 wherein the processor uses zero overhead to switch execution from one thread to the next.

20. The system according to claim 19 wherein each thread is given access to an array of general purpose registers and local data storage to enable switching with zero overhead.

21. The system according to claim 20 wherein the general purpose registers and the local data storage are made available to the processor by providing one address bit under the control of the thread execution control logic and by providing the remaining address bits under the control of the processor.

22. The system according to claim 20 wherein the processor is capable of simultaneously addressing multiple register arrays, and the thread execution control logic includes a selector to select which array will be delivered to the processor for a given thread.

23. The system according to claim 20 wherein the local data storage is fully addressable by the processor, an index register is contained within the register array, and the thread execution control has no address control over the local data storage or the register arrays.

24. A network processor configuration comprising:

a CPU with multiple threads;

an instruction memory, and a separate prefetch queue for each thread between the instruction memory and the CPU;

an array of general purposes registers, said array communicating with the CPU;

a local data storage including separate storage space for each thread;

a thread execution control for the general register array and the local data storage;

a first coprocessor connecting the CPU to a local data storage, said thread execution control including a priority FIFO buffer to store thread numbers;

a shared remote storage; and

a pipelined coprocessor connecting the shared remote storage and the CPU.

25. The processor configuration according to claim 24 wherein the thread execution control further includes a plurality of thread control state machines, one for each execution thread, and an arbiter responsive to signals from the FIFO buffer and the state machine to determine thread execution priority.

26. The use of prefetch buffers in connection with a plurality of independent instruction threads used to process data in a Network Processor comprising the steps of:

a) associating each thread with a prefetch buffer;

b) determining whether a buffer associated with an execution thread is full;

c) determining whether the thread associated with the buffer is active; and

d) during periods that the buffer is not being used by an active execution thread, enabling the buffer to prefetch instructions for the execution thread.

27. A thread execution control useful for the efficient execution of independent threads comprising:

a) a priority FIFO buffer for storing thread numbers;

b) a plurality of thread control state machines, one for each thread; and

c) an arbiter for determining the thread execution priority among multiple threads based upon signals outputted from the FIFO buffer and the state machines.

28. The thread execution control according to claim 27 wherein the FIFO includes :

a) means for loading a thread number into the FIFO when a packet is dispatched to the processor;

b) means for unloading a thread number from the FIFO when a packet has been enqueued for transmission;

(c) thread number transfer from highest priority to lowest priority in the FIFO when a long latency event occurs, and

(d) thread outlets of the FIFO used to determine priority depending on the length of time a thread has been in FIFO.

29. The thread execution control according to claim 27 wherein the arbiter controls the priority of execution of multiple independent threads based on the Boolean expression:

$$G_n = R_n \cdot \{(P_A = n) + R_{PA} \cdot (P_B = n) + R_{PA} \cdot R_{PB} \cdot (P_C = n) \cdots\}$$

where: G is a grant

R_n is a request from a given thread;

P_A, P_B and P_C represent threads ranked by alphabetical subscript according to priority;

$_n$ is a subscript identifying a thread by the bit or binary number comprising

a) determining whether a request R is active or inactive;

b) determining the priority of the threads;

c) matching the request R with the corresponding thread P; and

d) granting a request for execution if the request is active and if the corresponding thread P has the highest priority.

30. The thread execution control according to claim 27 wherein the thread control

state machine comprises control logic to:

(a) dispatch a packet to a thread;

(b) move the thread from an initialize state to a ready state;

(c) request execution cycles for the thread;

(d) move the thread to the execute state upon grant by the arbiter of an execution cycle;

(e) continue to request execution cycles while the thread is queued in the execute state; and

(f) return the thread to the initialize state if there is no latency event, or send the thread to the wait state upon occurrence of a latency event.

31. The thread executing control according to claim 28 wherein the FIFO further includes means to detect occurrence of latency events.